

IN THE CLAIMS:

1. (Currently Amended) A system comprising:

a data filter coupled to a text-to-image converter for converting filtered data into ~~a set of digital images, the set of digital images being defined as~~ image data using a set of texture maps; and

a memory analyzer for analyzing set-top box layout to determine types and corresponding sizes of memory available in the set-top box ~~and indicating available memory types~~, the memory analyzer being coupled to a memory distributor, wherein the system determines a storage size necessary for storing the texture maps defining one or more images and the memory distributor distributing distributes the texture maps to a preferred first memory if the total size of the texture maps is less than or equal to the available first memory size and further wherein if the total size of the texture maps is greater than the available first memory size, the texture maps being divided into two groups where a first group's total size is the largest possible sum of map sizes less than the first memory size and where a second group's total size is the difference between the total size of the set of texture maps and the total size of the first group.

2. (Original) The system of claim 1, wherein a total size of the set of texture maps is a sum of all texture map sizes.

3. (Currently Amended) The system of claim 2, further comprising:

a processor coupled to the data filter, wherein the processor ~~executing~~ executes a first logic in which the total size of the set of texture maps is less than or equal to ~~a~~ the first memory size; and

a second logic if the total size of the set of texture maps is greater than the first memory size~~[[,]]~~ ~~then dividing~~ to divide the set of texture maps into ~~at least~~ the two groups.

4. (Withdrawn) The system of claim 3, wherein a total size of the first group is the largest possible sum of texture map sizes for which the total size of the first group is less than the memory size.

5. (Withdrawn) The system of claim 3, wherein a total size of the second group is the difference between the total size of the set of texture maps and the total size of the first group.

6. (Currently Amended) The system of claim 3, wherein the set of texture maps of the first group is stored in a the first memory.

7. (Original) The system of claim 3, wherein the set of texture maps of the second group is stored in a second memory.

8. (Original) The system of claim 3, wherein the set of texture maps of the second group is compressed to fit into the first memory.

9. (Original) The system of claim 7, further comprising a compression engine.

10. (Currently Amended) A method comprising:

computing a total size of a set of texture maps;

searching for types of memories in a set-top box;

comparing the total size of the set of texture maps with a ~~memory~~ size of a preferred first memory;

dividing the set of texture maps into at least two groups if the total size of the set of texture maps is larger than the first memory size, such that the total size of the texture maps in a first group is the largest possible sum of texture map sizes for which the total size of texture maps is less than the first memory size and the total size of a second group is the difference between the total size of the set of texture maps and the total size of the first group

11. (Original) The method of claim 10 wherein computing a total size of a set of texture maps comprises:

computing a sum of all texture maps.

12. (Currently Amended) The method of claim 10 further comprising:

storing the set of texture maps in [[a]] the first memory if the total size of the set of texture maps is less than or equal to the first memory size.

13. (Currently Amended) The method of claim 10 further comprising:

storing [[a]] the first group of texture maps in [[a]] the first memory.

14. (Currently Amended) The method of claim 10 further comprising:

storing [[a]] the second group of texture maps in a second memory.

15. (Currently Amended) The method of claim 14 further comprising:

compressing the second group of texture maps to fit into [[C]] the first memory if

[[B]] the second memory is not available.